JavaScript – theory and tasks

Variables

To write programs in JavaScript, we'll need to make our code reusable.

Part of making code reusable is removing the data we want to perform some logic on to leave only the logic. Then we can use our logic on any data. Here's what we mean:

Imagine you're writing a weather app. Your thermometer outside reports the temperature in Celsius and your goal is to report the temperature in Fahrenheit.

You write a program that takes a temperature like *15* degrees Celsius, then multiplies and divides it until you get the temperature in Fahrenheit.

Once you've done this though, you see the temperature now reads *16* degrees Celsius. In order to find Fahrenheit again, you'd need to write a whole new program to convert *16* degrees Celsius to Fahrenheit.

This would happen because the Celsius-to-Fahrenheit logic is operating directly on the Celsius data.

Luckily, JavaScript offers variables. Variables allow us to assign data to a word, then we can use that word within our program instead of the data. Then, if the variable's value changes we only have to change the variable's value instead of re-writing the entire program.

Let's dive in and see a variable in the wild. Here is how you declare a variable:

var myName = 'Arya';

console.log(myName);

// Output: Arya

You can almost read it aloud: "Variable myName is equal to Arya."

Let's dissect that statement and look at its parts:

1. *var*, short for variable, is the JavaScript keyword that will create a new variable for us.
2. *myName* is chosen by a developer (that's you!). Notice that the word has no spaces, and each new word is capitalized. This is a common convention in JavaScript, and is called camelCase.
3. *=* means to assign whatever's next to the variable.
4. *'Arya'* is the value that the equals *=* assigns into the variable *myName*.

After the variable is declared, we can print the variable with: *console.log(myName)*. This will print *'Arya'* to the console.

***Instructions***

*Write your code in the createVariable.js file.*

1. Variables must be strong to hold all of our stuff. Exactly how strong are they?

Create a variable named strength, and assign the string '50,000 pounds' to it.

1. Now, under the strength variable, type

console.log('How much stuff can a variable hold? ', strength);

Now we know: What can a variable hold? Just about anything!

If variables can hold strings, can they hold other data types? Let's give it a shot:

var myAge = 15;

var likesChocolate = true;

console.log(myAge);

// Output: 15

console.log(likesChocolate);

// Output: true

Variables can hold any data type, like strings, numbers, and Booleans. They can also hold data types that we have not learned yet, like arrays, functions and objects (more on that later).

***Instructions***

*Write your code in the createVariable.js file.*

1. Under the strength variable, create a new variable named age and set it equal to your age as a number.
2. Under your age variable, use console.log to print it to the console.
3. Now, create another variable named hasPet, and set it equal to a boolean.

Then, print it to the console with console.log.

1. Way to go! Variables are able to hold any kind of data type.

Why do we care about variables?

Variables are useful in two ways:

1. They allow us to use the same value over and over, without having to write a string or other data type over and over.
2. More importantly, we can assign variables different values that can be read and changed by the program without altering our code.

For example, a weather app can show you a different high temperature every day. Instead of writing a new website everyday, they store the information in a variable and just change the value of that variable.

We can change a variable's value if we want, like this:

var weatherCondition = 'Monday: Raining cats and dogs';

weatherCondition = 'Tuesday: Sunny';

console.log(weatherCondition);

// Output: 'Tuesday: Sunny'

1. We created a variable by using the keyword *var* and the name *weatherCondition*.
2. Then, we took the existing *weatherCondition* variable, and set its value equal to *'Tuesday: Sunny'*.

True to their name, variables are indeed variable.

***Instructions***

*Write your code in the changeVariable.js file.*

1. Create a variable on line 1 named morningAlarm and set it to '6:30AM'.
2. On line 2, set morningAlarm equal to another time that's better for you, that is not at 6:30AM.
3. On line 3, use

console.log('Morning alarm is set to: ', morningAlarm) to print out your alarm to the console.

Notice that you successfully changed the variable's value because of your code on line 2.

In the previous lessons, we've put strings into variables. Now, let's put a variable's value into a string!

Putting a variable in a string uses concepts we've already learned. The JavaScript term for this idea is interpolation.

Interpolwhat?! —Possibly the most fun JavaScript term to say.

We can use the + operator from earlier to interpolate (insert) a variable into a string, like this:

var myPet = 'armadillo';

console.log('I own a pet ' + myPet + '.');

// Output: 'I own a pet armadillo.'

***Instructions***

*Write your code in the stringInterpolation.js file.*

1. Create a variable named favoriteAnimal and set it equal to your favorite animal.
2. Then, use console.log to print: 'My favorite animal: Koala', but replace 'Koala' with your favoriteAnimal variable.

You made it to the end of the first unit and this lesson! High five!

We learned:

* How to create variables.
* How to change a variable's value.
* How to interpolate, or insert, a variable into a string.

In the next lesson, we will learn how to program JavaScript to make decisions for us and how to generate random numbers.